**Bài tập thực hành 04**

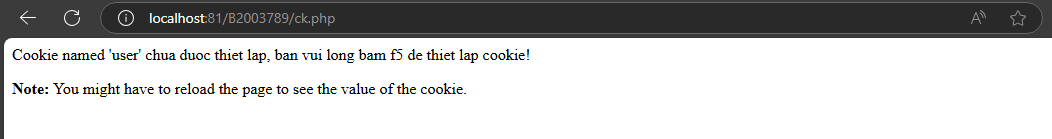
**Họ tên: Trần Anh Khoa**

**MSSV : B2003789**

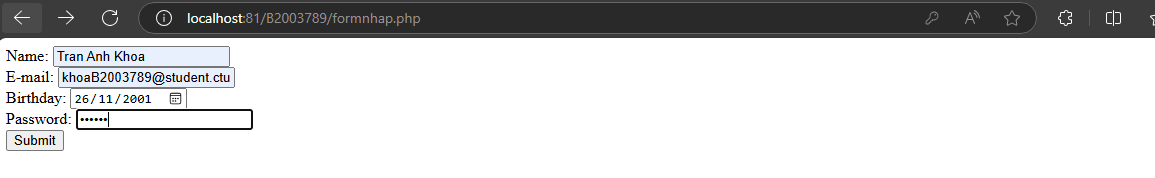
**Lập trình PHP nâng cao**

Câu 1 )

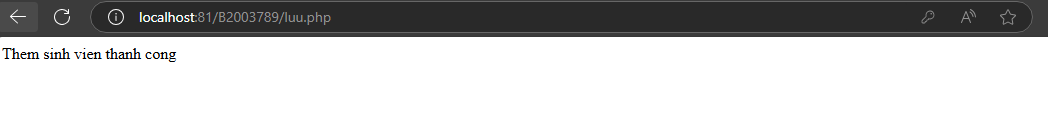
ck.php



Formnhap.php



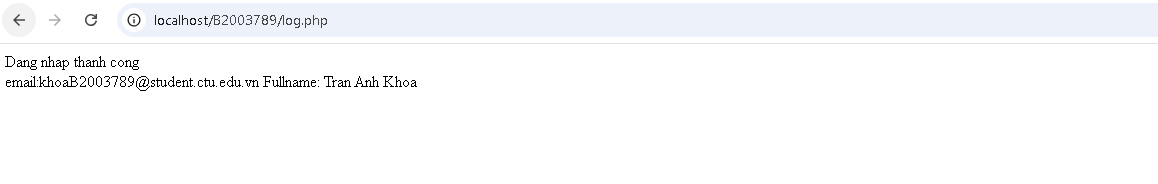
Luu.php



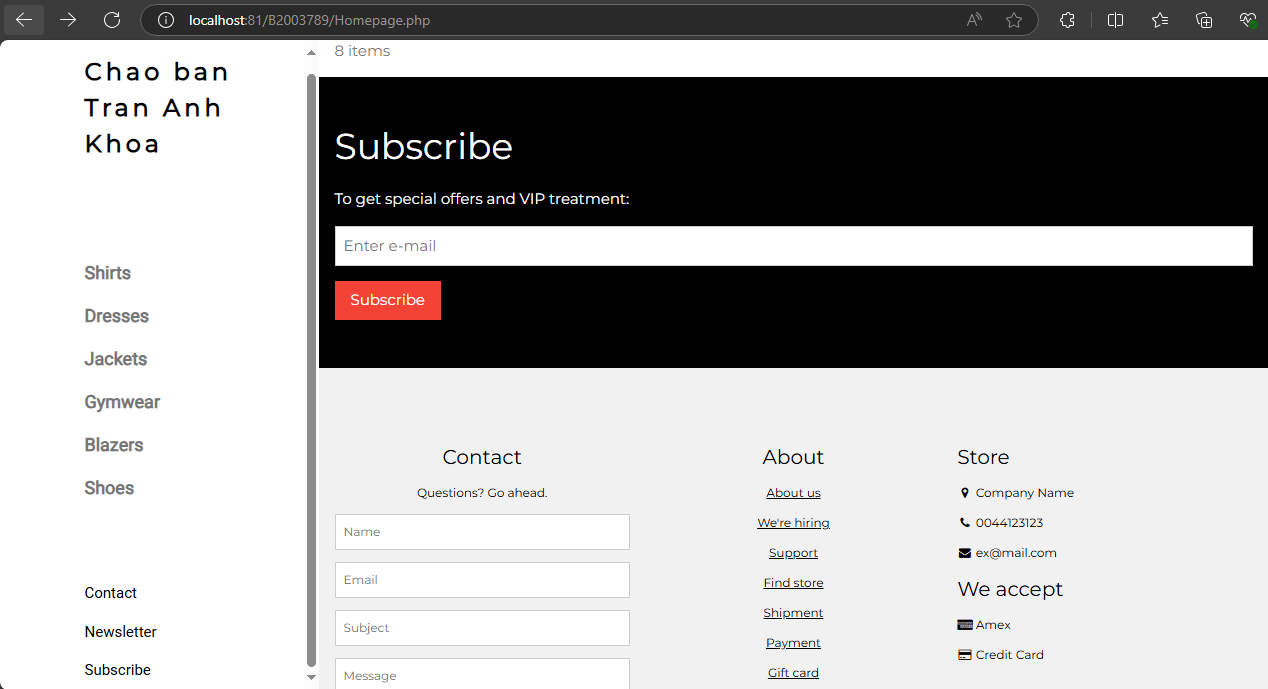
Login.php



Log.php



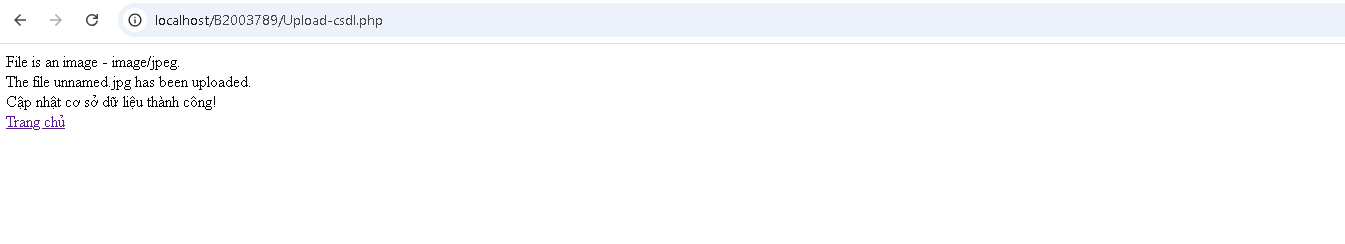
Homepage.php



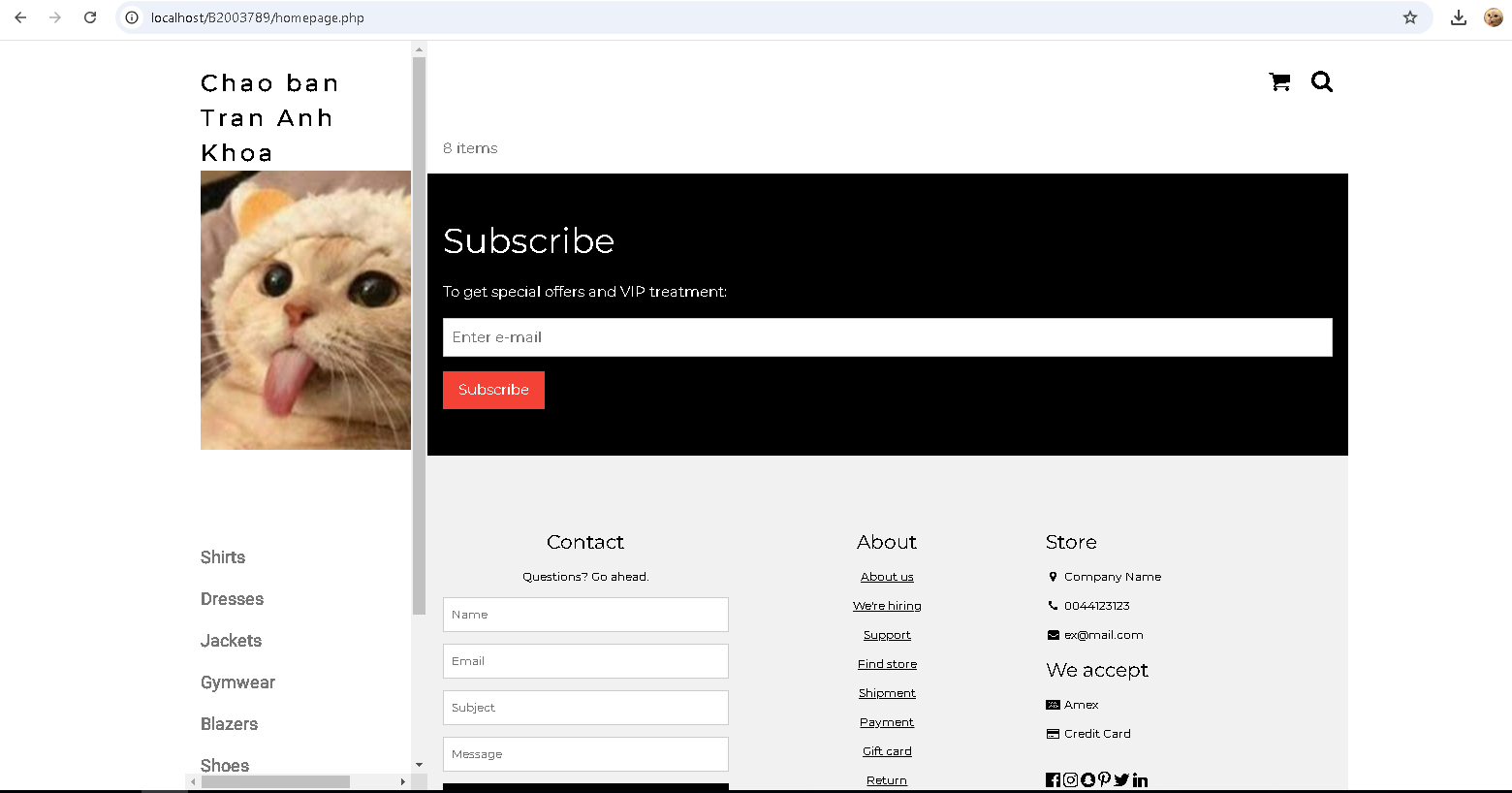
Upload-img.php



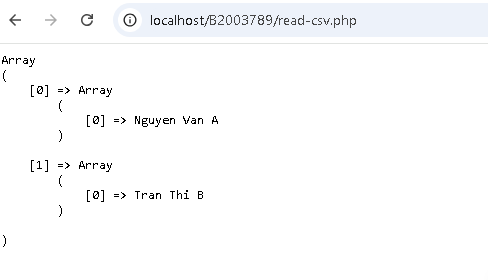
Upload-csdl.php



Homepage.php



Read-csv.php



**Câu 2 )Trong vài trường hợp, hacker có thể sử dụng các kỹ thuật SQL Injection để hack hệ thống của bạn. Bạn hãy trình bày SQL Injection là gì và thử các ví dụ trình bày trong 1. Ứng dụng kỹ thuật đó vào trang đăng nhập mà bạn đã tạo, chụp lại các kết quả.**

**SQL Injection** là một kỹ thuật tấn công bảo mật mà kẻ tấn công chèn hoặc "inject" các mã SQL độc hại vào các truy vấn SQL của ứng dụng web thông qua đầu vào của người dùng. Mục đích của SQL Injection có thể là:

* **Lấy cắp dữ liệu**: Truy cập dữ liệu nhạy cảm trong cơ sở dữ liệu.
* **Chỉnh sửa dữ liệu**: Thay đổi hoặc xóa dữ liệu.
* **Xóa dữ liệu**: Xóa hoàn toàn dữ liệu trong cơ sở dữ liệu.
* **Thực thi mã**: Thực thi các câu lệnh SQL hoặc mã hệ thống khác, có thể dẫn đến việc tấn công máy chủ.

**Câu 3) Dựa vào link2 , bạn hãy cho biết Cookie là gì, diễn giải ý nghĩa các tham số trong setcookie(). Cách lưu, lấy, xóa giá trị trong cookie.**

**Cookie** là một tập tin dữ liệu nhỏ được lưu trữ trên máy tính của người dùng bởi trình duyệt web khi người dùng truy cập vào một trang web. Cookies được sử dụng để lưu trữ thông tin về người dùng và các trạng thái của trang web giữa các phiên làm việc, giúp cải thiện trải nghiệm người dùng và duy trì trạng thái của ứng dụng.

**Ý Nghĩa Các Tham Số Trong setcookie()**

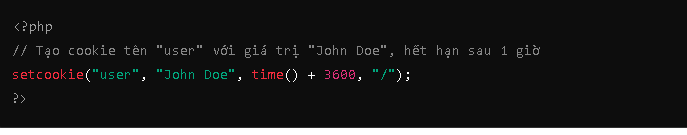
Hàm setcookie() trong PHP được sử dụng để tạo hoặc thay đổi một cookie. Cú pháp cơ bản của hàm setcookie() là:

bool setcookie(string $name [, string $value = "" [, int $expires = 0 [, string $path = "" [, string $domain = "" [, bool $secure = false [, bool $httponly = false]]]]])

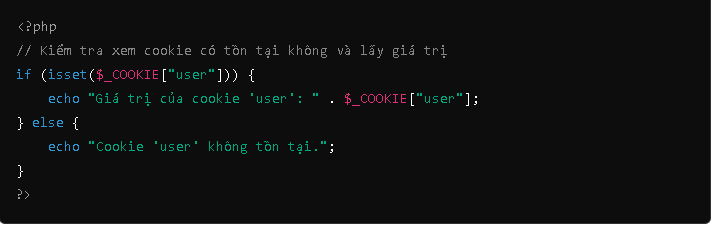
**Các tham số của setcookie() là:**

1. **$name**: Tên của cookie. Đây là tham số bắt buộc.
2. **$value**: Giá trị của cookie. Đây là tham số tùy chọn và có giá trị mặc định là chuỗi rỗng. Đây là giá trị sẽ được lưu trữ trong cookie và có thể là bất kỳ dữ liệu nào.
3. **$expires**: Thời gian hết hạn của cookie tính bằng giây từ thời điểm hiện tại. Nếu bạn đặt giá trị này là 0 (hoặc để trống), cookie sẽ hết hạn khi trình duyệt của người dùng đóng. Bạn có thể tính toán thời gian hết hạn bằng cách sử dụng hàm time() cộng với số giây cần thiết. Ví dụ: time() + 3600 để cookie hết hạn sau 1 giờ.
4. **$path**: Đường dẫn mà cookie có hiệu lực. Nếu để trống, cookie sẽ chỉ có hiệu lực trong thư mục của trang đã gửi cookie. Ví dụ: / để cookie có hiệu lực trên toàn bộ trang web.
5. **$domain**: Tên miền mà cookie có hiệu lực. Nếu để trống, cookie sẽ chỉ có hiệu lực trên tên miền hiện tại. Bạn có thể đặt giá trị này thành example.com để cookie có hiệu lực trên toàn bộ miền example.com và các tên miền con.
6. **$secure**: Nếu tham số này được đặt thành true, cookie chỉ được gửi qua kết nối HTTPS (SSL). Nếu để false, cookie có thể được gửi qua kết nối không bảo mật.
7. **$httponly**: Nếu tham số này được đặt thành true, cookie chỉ có thể được truy cập qua giao thức HTTP và không thể được truy cập qua JavaScript. Điều này giúp bảo vệ cookie khỏi các cuộc tấn công XSS (Cross-Site Scripting). Nếu để false, cookie có thể được truy cập qua cả JavaScript và HTTP.

Tạo cookie :



Lấy cookie:



Xóa cookie:



**Câu 4. Dựa vào link3, bạn hãy cho biết Session dùng để làm gì. Cách lưu, lấy, xóa giá trị trong Session**.

**Session** là một cơ chế lưu trữ dữ liệu tạm thời trên máy chủ để theo dõi thông tin của người dùng trong suốt một phiên làm việc. Nó cho phép bạn lưu trữ thông tin về người dùng giữa các yêu cầu HTTP, điều này rất quan trọng trong các ứng dụng web yêu cầu quản lý trạng thái, như các trang web đăng nhập và giỏ hàng.

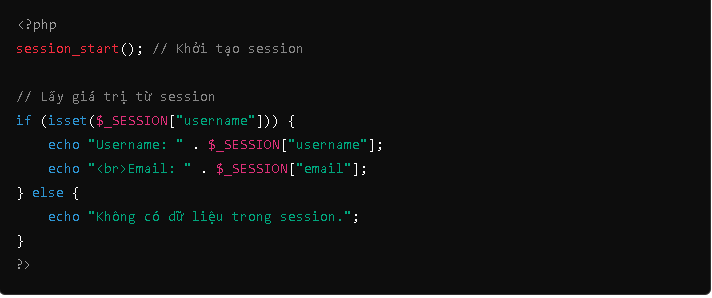
**Ý Nghĩa của Session**

* **Theo Dõi Người Dùng**: Giúp theo dõi trạng thái và dữ liệu của người dùng giữa các trang và các yêu cầu.
* **Lưu Trữ Dữ Liệu Tạm Thời**: Dữ liệu lưu trữ trong session có thể bao gồm thông tin người dùng, giỏ hàng, các tùy chọn cấu hình, v.v.
* **Bảo Mật**: Session được lưu trữ trên máy chủ, giúp bảo vệ dữ liệu nhạy cảm khỏi các cuộc tấn công từ bên ngoài.

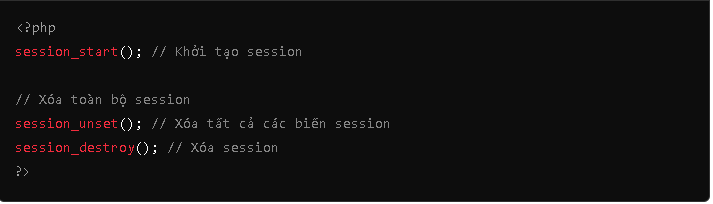
**Lưu giá trị Session:**



**Lấy giá trị Session:**



**Xóa Session:**



**Câu 5 . Bạn hãy so sánh Cookie và Session.**

 **Lưu Trữ**:

* **Cookie**: Được lưu trữ trên máy khách (trình duyệt của người dùng). Điều này có nghĩa là mỗi khi người dùng gửi yêu cầu đến máy chủ, cookie sẽ được gửi kèm theo yêu cầu.
* **Session**: Được lưu trữ trên máy chủ. Trình duyệt của người dùng chỉ lưu trữ một ID phiên (session ID) trong cookie, và tất cả dữ liệu phiên được lưu trữ trên máy chủ.

 **Kích Thước**:

* **Cookie**: Thông thường có giới hạn kích thước khoảng 4KB cho mỗi cookie. Nếu bạn cần lưu trữ nhiều dữ liệu, bạn có thể cần phải sử dụng nhiều cookie hoặc lưu trữ dữ liệu lớn hơn ở nơi khác.
* **Session**: Không bị giới hạn kích thước dữ liệu như cookie. Dữ liệu có thể lớn hơn vì nó được lưu trữ trên máy chủ.

 **Thời Gian Hết Hạn**:

* **Cookie**: Có thể thiết lập thời gian hết hạn cụ thể, từ vài giây đến nhiều năm. Cookie sẽ tự động hết hạn khi thời gian này đã qua.
* **Session**: Thường hết hạn khi người dùng đóng trình duyệt hoặc sau một khoảng thời gian không hoạt động. Thời gian hết hạn có thể được cấu hình trên máy chủ.

 **Bảo Mật**:

* **Cookie**: Dữ liệu trong cookie có thể bị truy cập và sửa đổi bởi người dùng thông qua các công cụ phát triển trình duyệt. Cần sử dụng các thuộc tính Secure và HttpOnly để bảo vệ dữ liệu cookie.
* **Session**: Dữ liệu chỉ được lưu trữ trên máy chủ và không thể bị truy cập trực tiếp từ trình duyệt, giúp bảo vệ thông tin nhạy cảm tốt hơn.

 **Khả Năng Lưu Trữ**:

* **Cookie**: Dữ liệu lưu trữ trong cookie có thể tồn tại lâu dài, không phụ thuộc vào thời gian phiên làm việc của người dùng.
* **Session**: Dữ liệu tồn tại chỉ trong suốt thời gian phiên làm việc hoặc cho đến khi phiên làm việc bị hủy.

 **Truy Cập**:

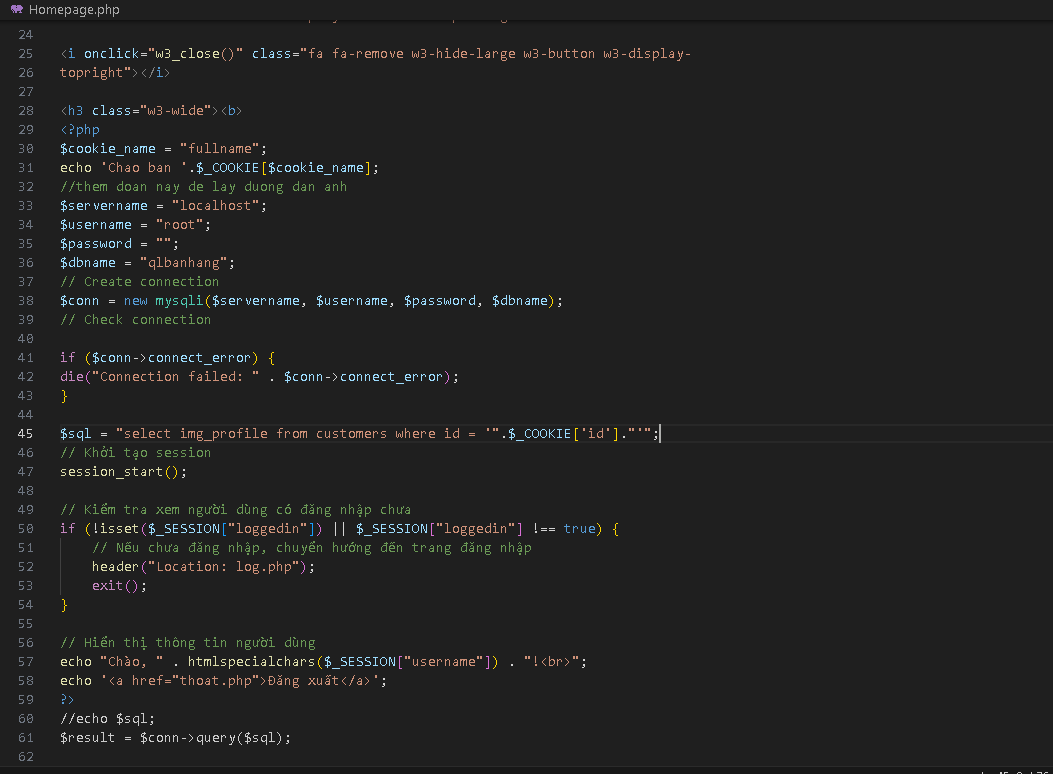
* **Cookie**: Có thể được truy cập từ JavaScript trên trình duyệt (trừ khi HttpOnly được thiết lập).
* **Session**: Không thể được truy cập từ JavaScript và chỉ có thể được truy cập thông qua các yêu cầu từ phía máy chủ.

**Câu 6) Chỉnh sửa các đoạn gán, khởi tạo, lấy giá trị cookie trong log.php và homepage.php thay bằng cách dùng Session. Tạo tập tin thoat.php để xóa các giá trị trong session, cookie thực hiện chức năng log out khỏi hệ thống.**

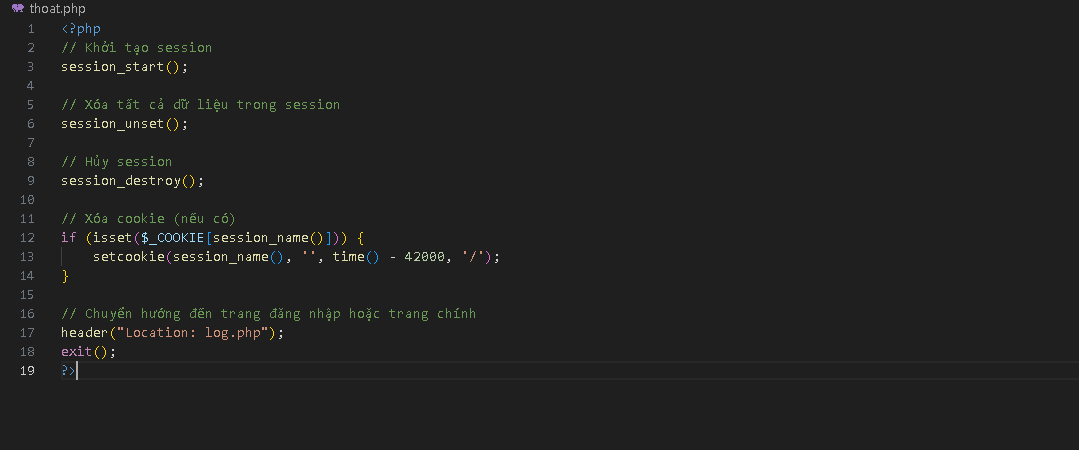
**Log.php**



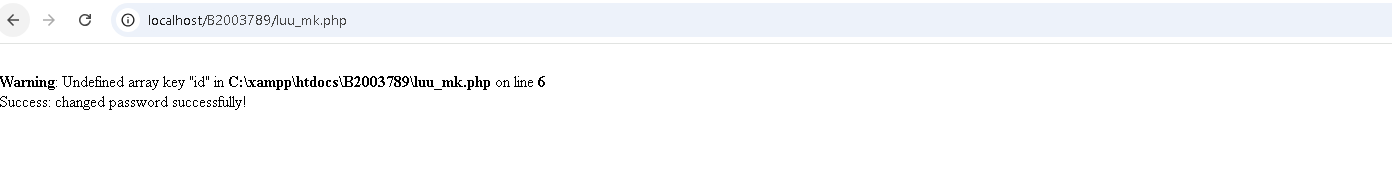
**Homepage.php**



**Thoat.php**

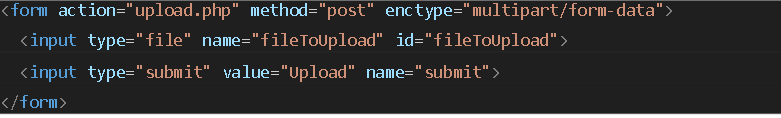


**Câu 7)Bạn hãy tạo form sua\_mk.php cho phép người dùng chỉnh sửa mật khẩu sau khi đăngnhập. Yêu cầu gồm 3 input: ô để nhập mật khẩu cũ, 1 ô để nhập mật khẩu mới, 1 ô chophép nhập mật lại mật khẩu mới. Khi nhập xong cần đảm bảo: mật khẩu cũ là khớp vớiCSDL đang lưu, 2 ô nhập mật khẩu mới phải khớp với nhau và không giống với mật khẩucũ, nếu đáp ứng điều kiện thì tiến hành băm mật khẩu với md5 và lưu mật khẩu mới vào CSDL.**

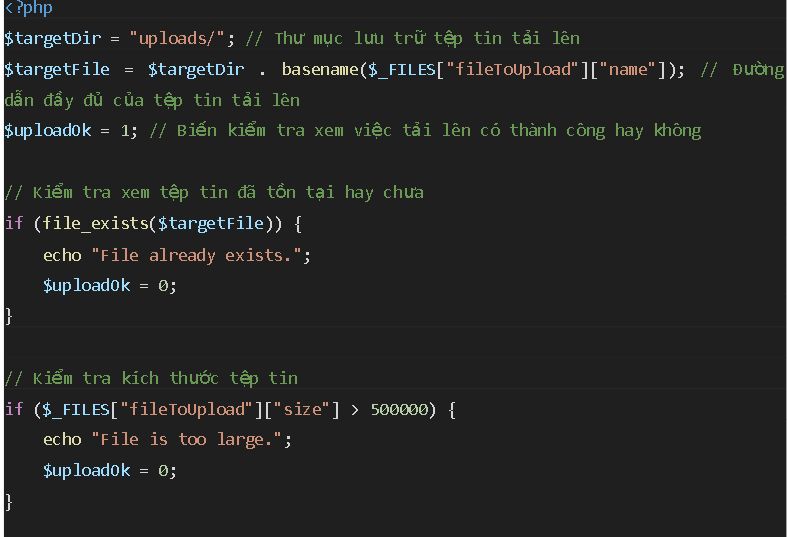


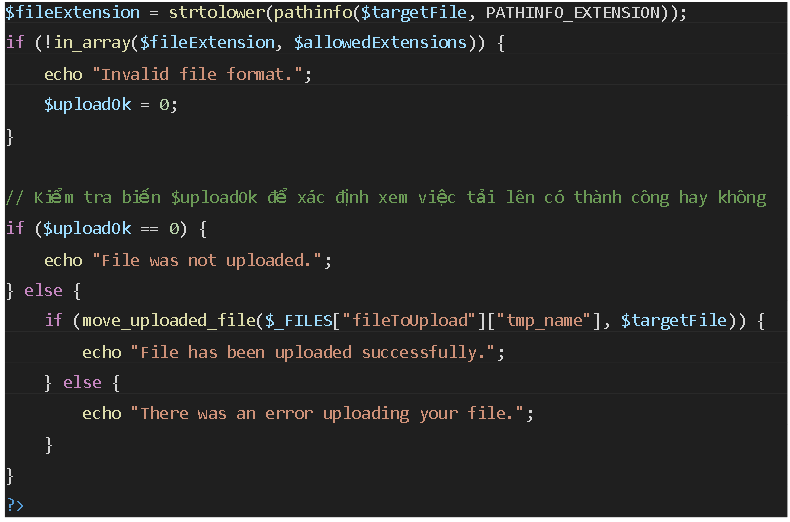
**Câu 8. Bạn đọc trong4 để tìm hiểu và mô tả các cơ chế, các hàm/thủ tục để thực hiện việc upload.**

Bước 1: Định nghĩa biểu mẫu HTML cho việc tải tập tin lên



Bước 2:Tạo tập tin PHP(ví dụ:upload.php) để xử lý việc tải lên





Trong ví dụ trên chúng ta sử dụng biến siêu toàn cục ‘$\_FILES’ để truy cập thông tin về tệp tin đã tải lên. ‘$\_FILES[“fileToUpload”][“name”] là tên tệp tin gốc

Bước 3 : Kiểm tra và xử lý lỗi khi tải tệp tin lên:

Sử dụng các câu lệnh điều kiện để kiểm tra và xử lý lỗi trong tập tin upload.php. Ví dụ : kiểm tra xem tệp tin đã tồn tại , kiểm tra kích thước và định dạng tệp tin , kiểm tra xem việc di chuyển tệp tin đã thành công hay không.

**Câu 9 Bạn hãy đọc, mô tả chức năng của hàm này và các tham số trong hàm (tham khảo thêm từ read-csv.php).**

fgetcsv() được sử dụng để đọc dữ liệu từ tệp CSV và phân tích dữ liệu đó thành mảng. Hàm này sẽ đọc một dòng từ tệp CSV và trả về một mảng chứa các trường dữ liệu từ dòng đó.

**Tham Số**:

1. **$handle**:
   * **Loại**: resource
   * **Mô Tả**: Đây là con trỏ tệp mở mà bạn muốn đọc từ đó. Con trỏ này được tạo ra bằng cách sử dụng hàm fopen().
   * **Ví Dụ**: fopen("file.csv", "r")
2. **$length** (Tùy chọn):
   * **Loại**: int
   * **Mô Tả**: Số lượng ký tự tối đa mà hàm sẽ đọc trong một lần. Nếu length được đặt thành 0, hàm sẽ đọc toàn bộ dòng cho đến khi gặp ký tự mới dòng hoặc kết thúc tệp.
   * **Giá Trị Mặc Định**: 0
3. **$delimiter** (Tùy chọn):
   * **Loại**: string
   * **Mô Tả**: Ký tự phân cách giữa các trường trong tệp CSV. Mặc định là dấu phẩy (,).
   * **Giá Trị Mặc Định**: ","
4. **$enclosure** (Tùy chọn):
   * **Loại**: string
   * **Mô Tả**: Ký tự bao quanh các trường trong tệp CSV. Mặc định là dấu nháy kép ("). Ví dụ, nếu một trường chứa dấu phân cách (,) bên trong nó, trường đó sẽ được bao quanh bởi ký tự enclosure.
   * **Giá Trị Mặc Định**: "\""
5. **$escape** (Tùy chọn):
   * **Loại**: string
   * **Mô Tả**: Ký tự được sử dụng để thoát ký tự enclosure bên trong các trường. Mặc định là dấu gạch chéo ngược (\).
   * **Giá Trị Mặc Định**: "\""

**Trả Về**:

* **Loại**: array hoặc false
* **Mô Tả**: Hàm trả về một mảng chứa các trường dữ liệu từ dòng được đọc. Nếu không còn dữ liệu để đọc (cuối tệp), hàm trả về false



**10. Bạn hãy tạo 1 tập tin csv ít nhất 10 dòng dữ liệu với các cột dữ liệu như bảng customer trong csdl qlbanhang. Tạo tập tin upload-csv.php với giao diện cho phép upload các tập**